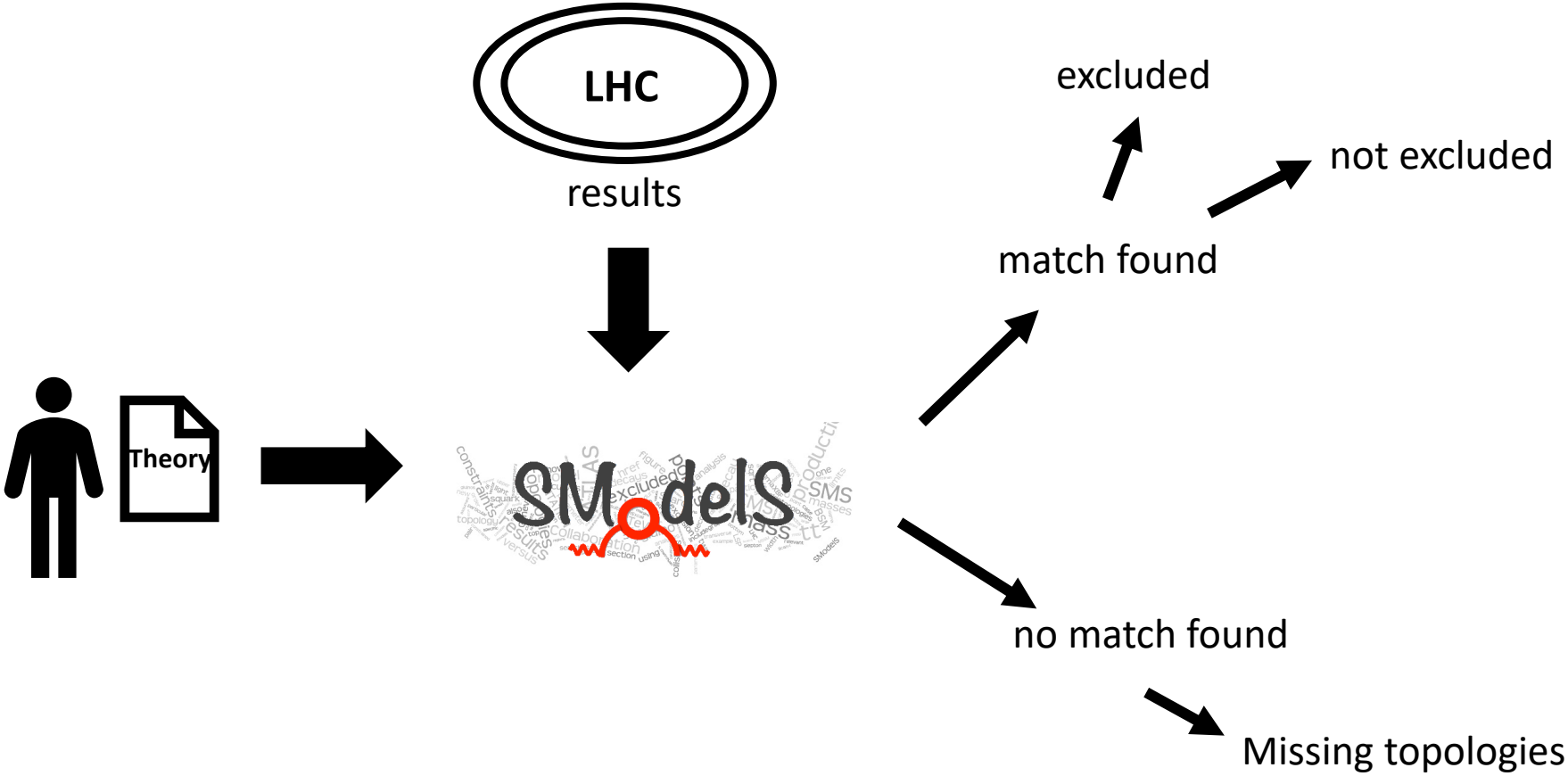
Development towards signatures beyond missing energy

ALICIA WONGEL

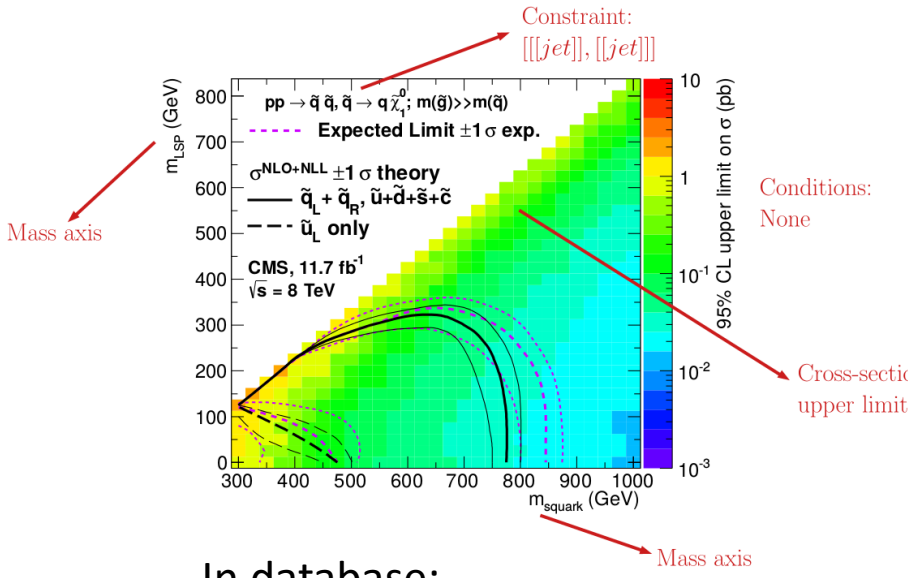
HEPHY
INSTITUT FÜR HOCHENERGIEPHYSIK

SModelS

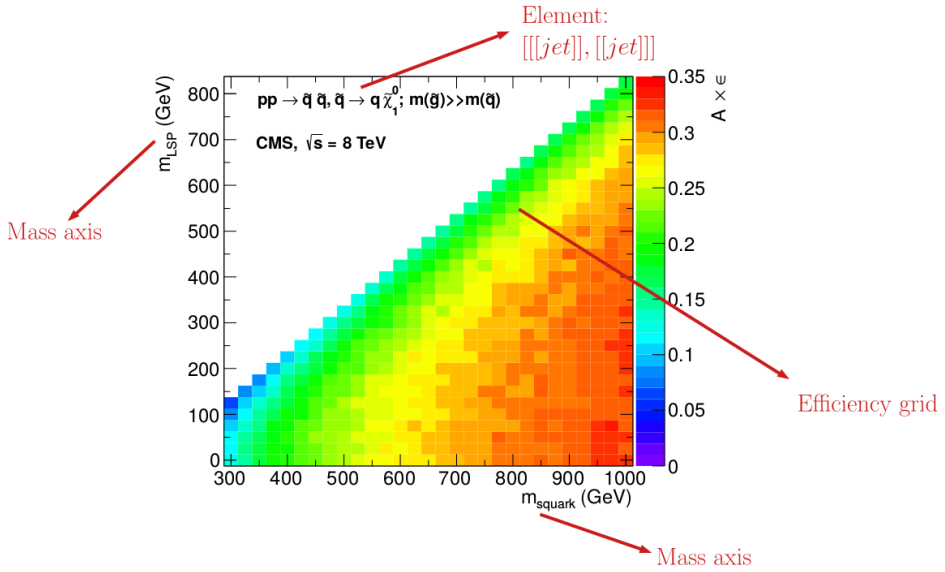

LHC
results
Theory
SModelS
excluded
not excluded
match found
no match found
Missing topologies

Simplified model results

Upper Limit



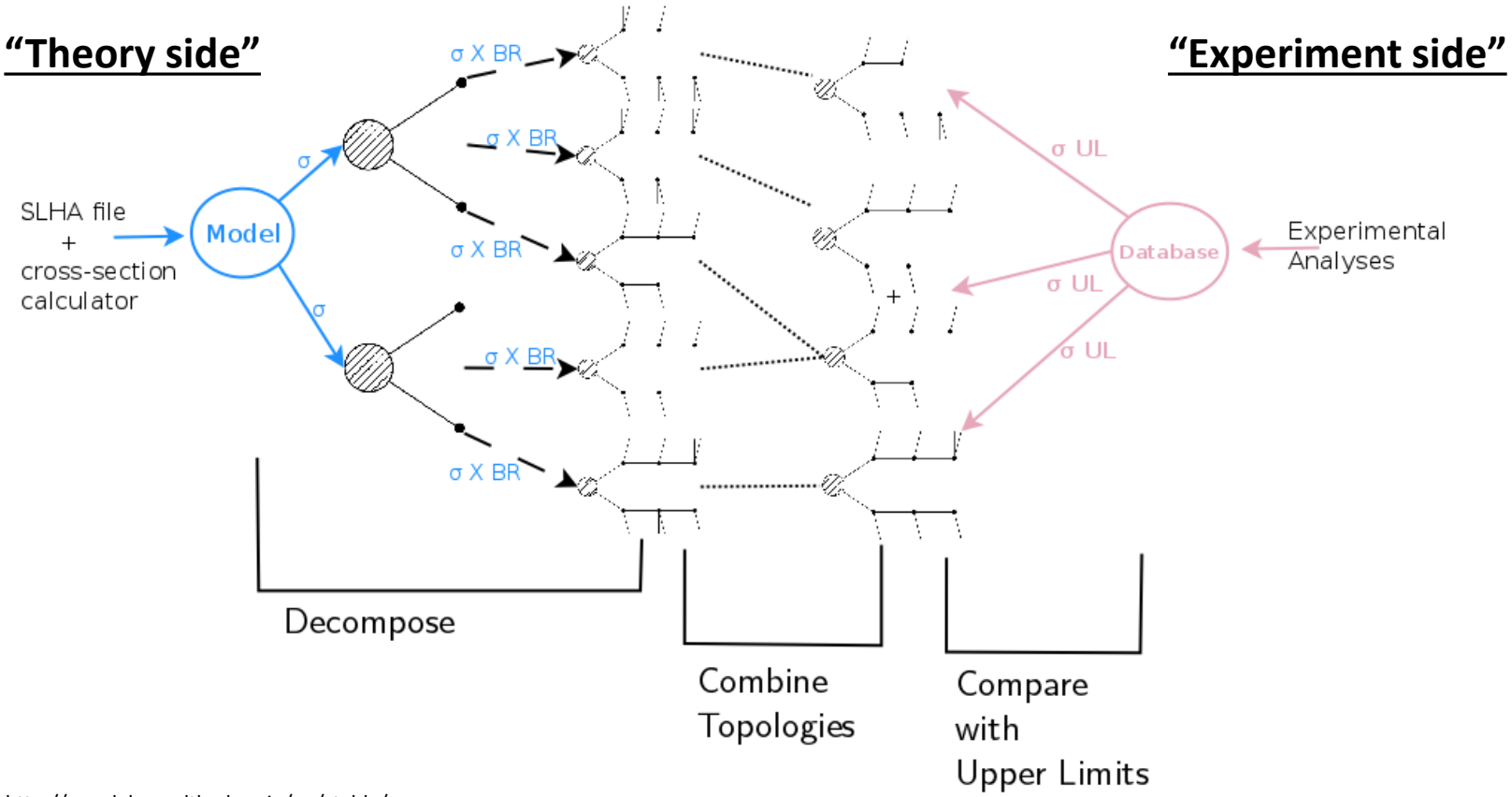
Efficiency Map



In database:

[[[m1, m2], [m1, m2]], σ] or [[[m1,m2], [m1,m2]], ε]

SModelS

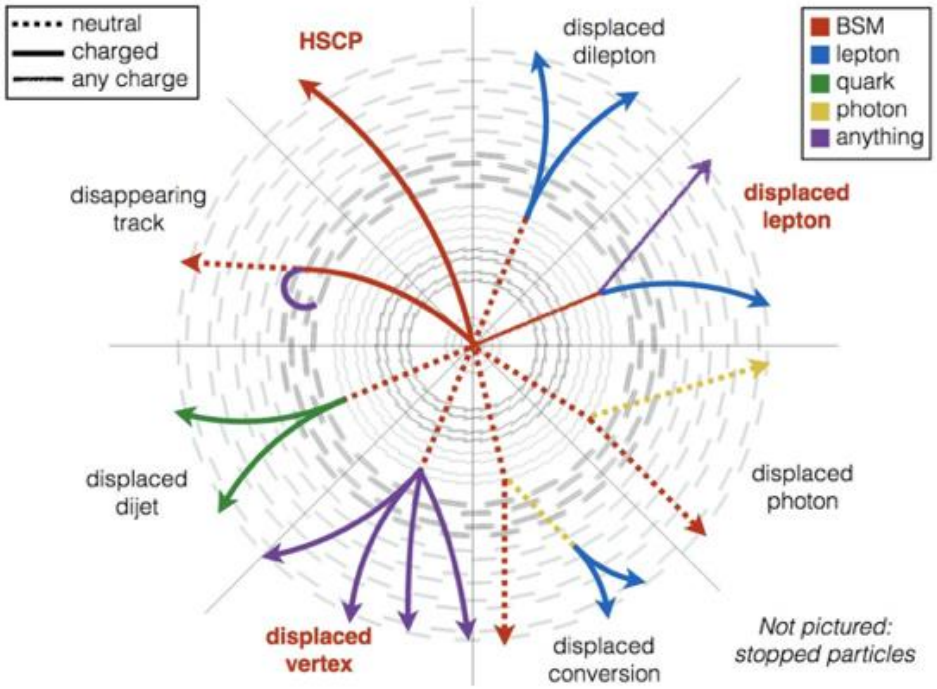


http://smodels.readthedocs.io/en/stable/

Existing framework

- Restricted to models with Z_2 - symmetry
- Characterize SM particles with string, BSM particles with string and mass
- Missing energy (MET) final states only + Heavy Stable Charged Particle (HSCP) in v1.2.0

Motivation



- No excess in MET final states
- Variety of BSM models feature beyond MET signature: e.g. SUSY with long lived chargino
- Need to keep track of quantum numbers, life time and mass of BSM particles

Image taken from talk by J.Antonelli at ICHEP 2016

Changes in the code

Particle class

SM example:

```
e = Particle(
Z2parity='even‘,
label='e-‘,
pdg=11,
mass=0.5*MeV,
eCharge=-1,
colordim=0,
spin=1./2,
totalwidth = 0.*GeV,
decays=[])
```

BSM example:

```
gluino =
Particle(
Z2parity='odd',
label='gluino',
pdg=1000021,
eCharge=0,
colordim=8,
spin=1./2)
```

- For BSM added from file: mass, width, decays

- → Branches also consist of objects
- → Elements also consist of objects

Experimental side

- Inclusive result (already in v1.2.0): [[*], [ ]]

- Final state particles:

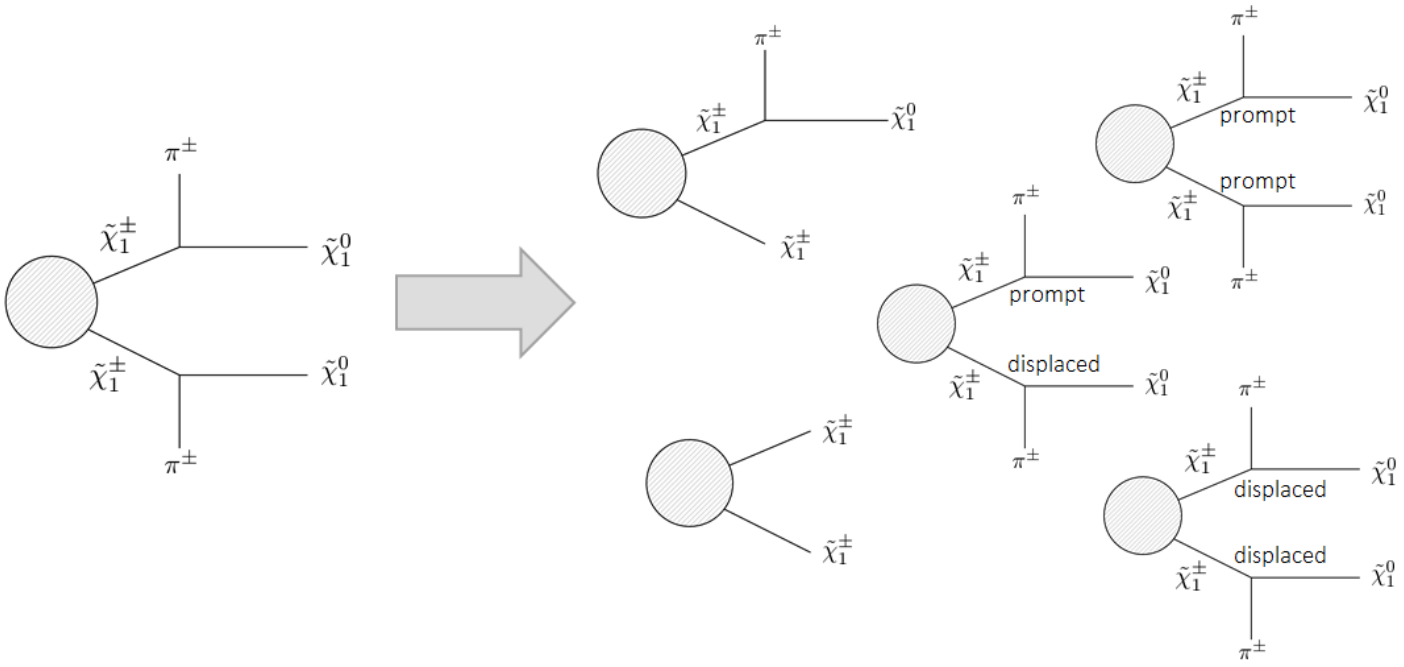
```
MET = Particle(
label='MET',
Z2parity ='odd',
eCharge = 0,
colordim = 0)
```

```
HSCPp = Particle(
label='HSCP+',
Z2parity ='odd',
eCharge = +1,
colordim = 0)
```

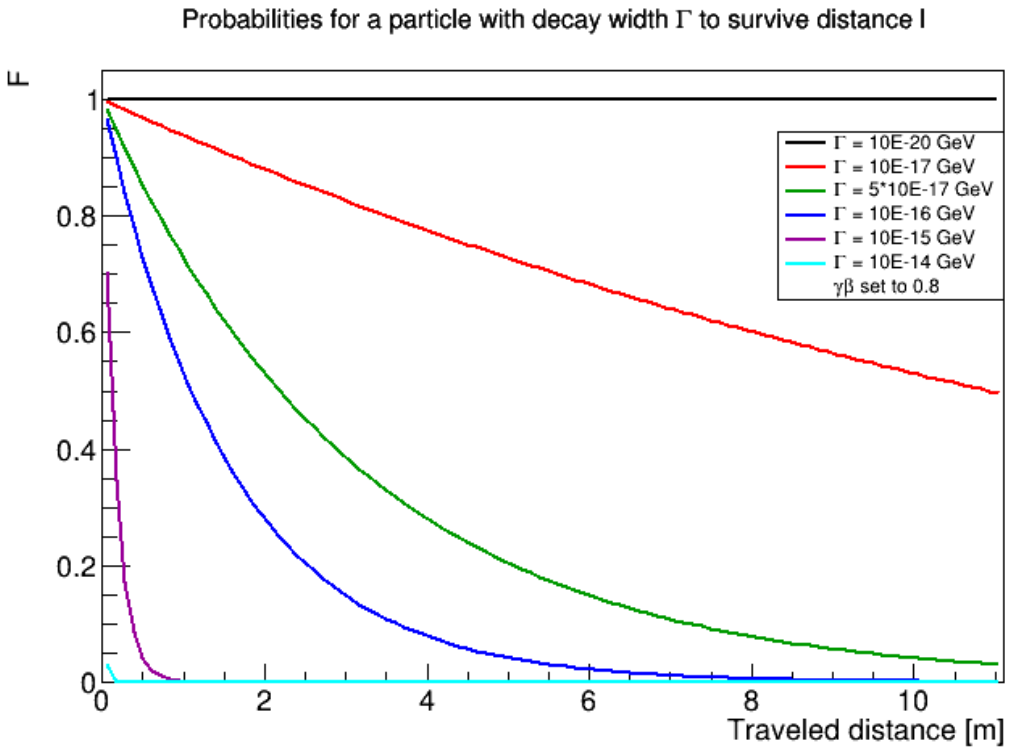
Challenges/ Ongoing work

Number of created elements

Large increasement of number of elements when signatures beyond missing energy are allowed



Decay probabilities



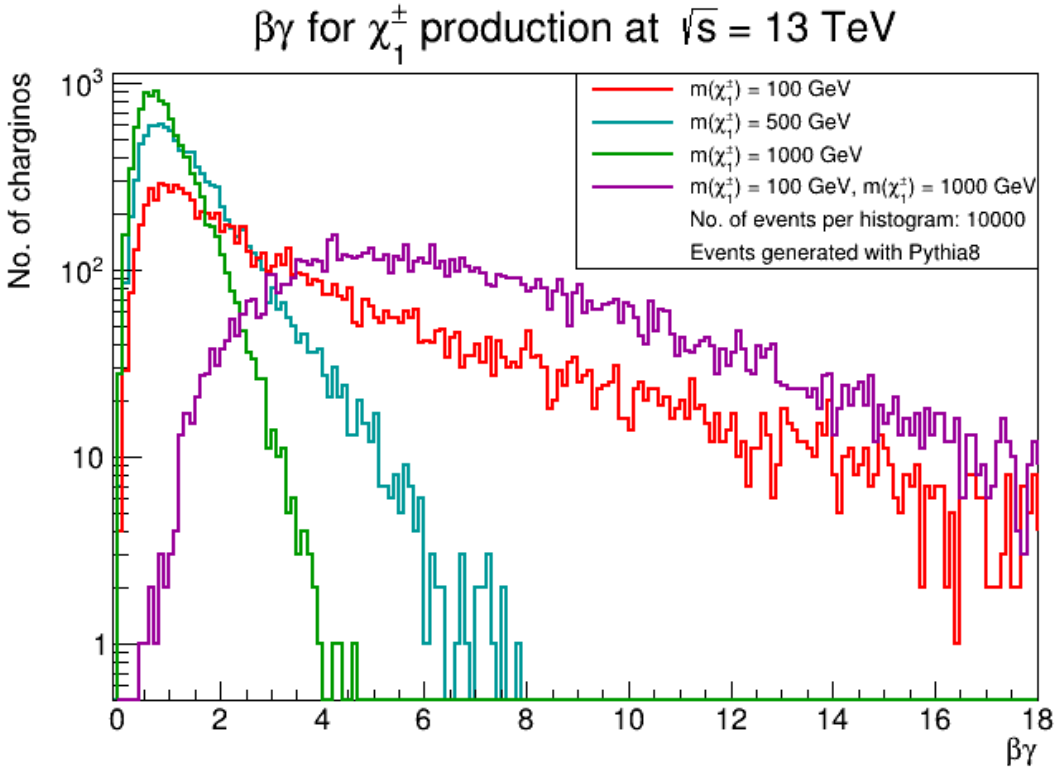
- Every particle (with a non-zero width) has probability to survive a certain distance

$$F = exp(-\Gamma \frac{l}{\beta\gamma\hbar c})$$

- βγ ? l ?

- In v.1.2.0 estimated βγ

βγ distribution



red, green, blue:

$pp \to \tilde{\chi}_1^\pm \tilde{\chi}_1^\pm$

purple:

$pp \to \tilde{\chi}_1^\pm \tilde{\chi}_2^0,$

$\tilde{\chi}_2^0 \to \tilde{\chi}_1^\pm \; soft \; SM$

→ Additional file in database with:

[[[m1],[m1]] , [[t1], [t1]] , F]

Summary

- Upgrade SModelS to include signatures beyond missing energies
- Particle class: particle properties, final states

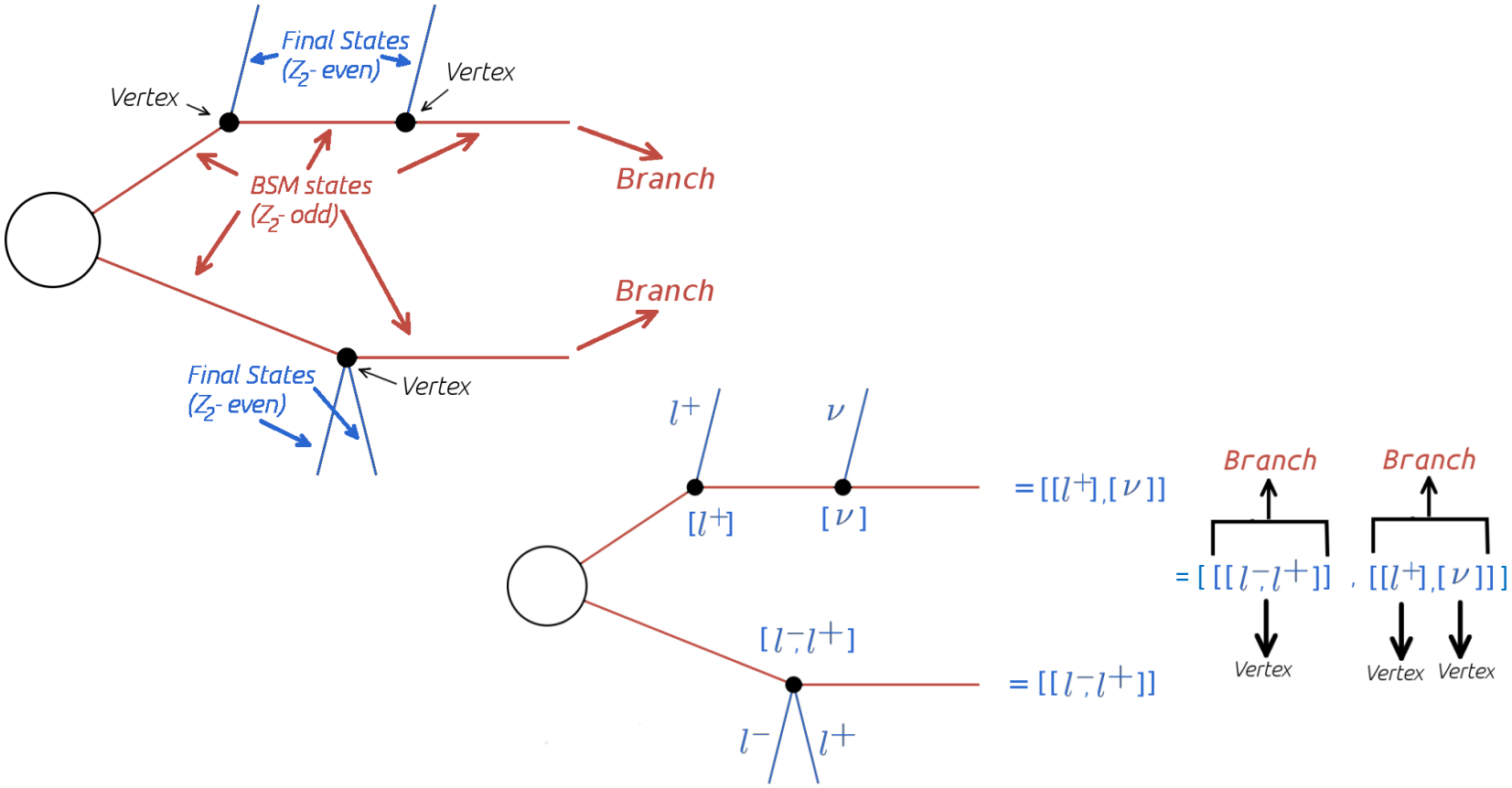
  → Particles' properties from file, long-lived particles added in decomposition
- Database: final states, inclusive results
- Reweighting of the cross section
  - Exponential function
  - Introduce into database
- Outlook: implement a displaced search

Backup

Simplified Models

- Involves few particles and decay modes
- Enhances applicability of new physics searches
- Often limits of more general new physics scenarios
- Mainly used to identify limits

Terminology



http://smodels.readthedocs.io/en/stable/TheoryDefinitions.html

Model

Get masses and decays from file

- `lheReader`: get masses and decays from lhe file
- For slha file masses and decays can be read off directly, additionally width

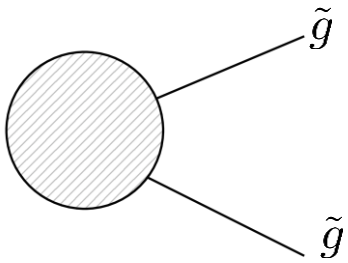
Update particles

Decompose model instead of file

Instead of lhe/slha decomposer just one decomposer

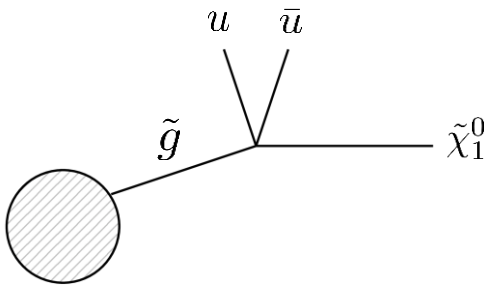
Decomposition

Production cross section from model

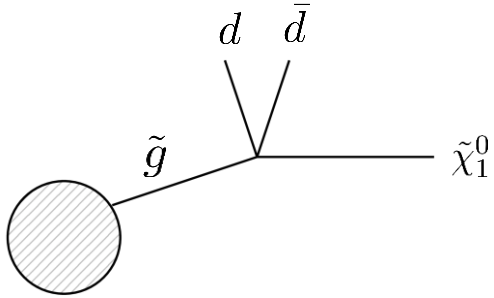


σ

Read decay(s) from model:
What decays does the gluino have? [0.5[1,-1,1000022], 0.5[2, -2,1000022]]
Add new branch for each decay



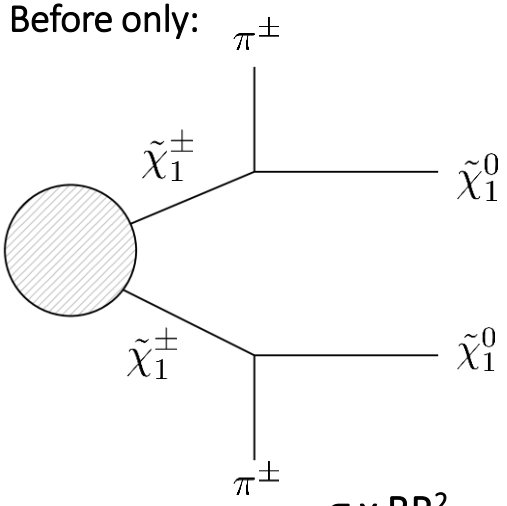
σ x BR1



σ x BR2

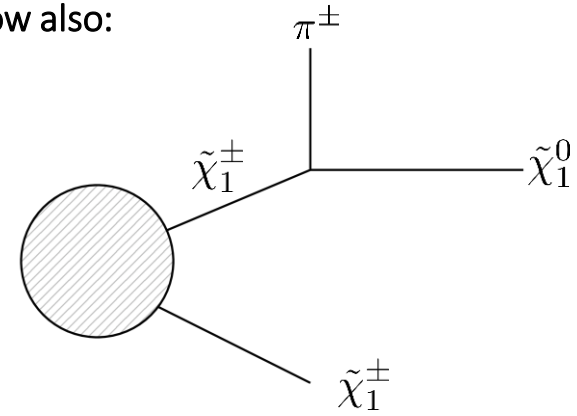
Decomposition for long-lived particles (already in v1.2.0)

Before only:



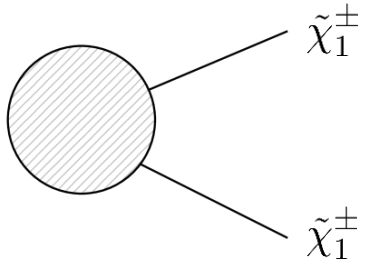
$\sigma \times BR^2$

Now also:



$\sigma \times BR$

and:



σ

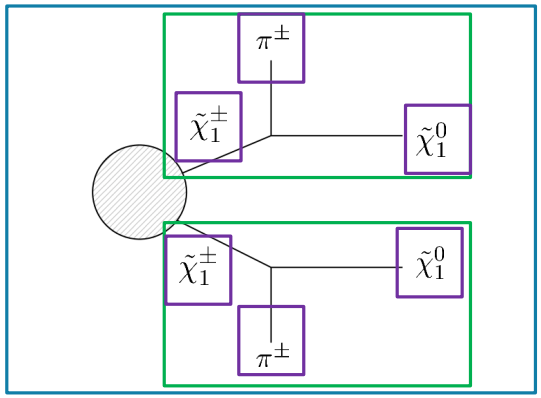
Consequences of particle class

- Particles now `Particle class` objects
- Branches also contain `Particle class` objects
- Elements also contain `Particle class` objects

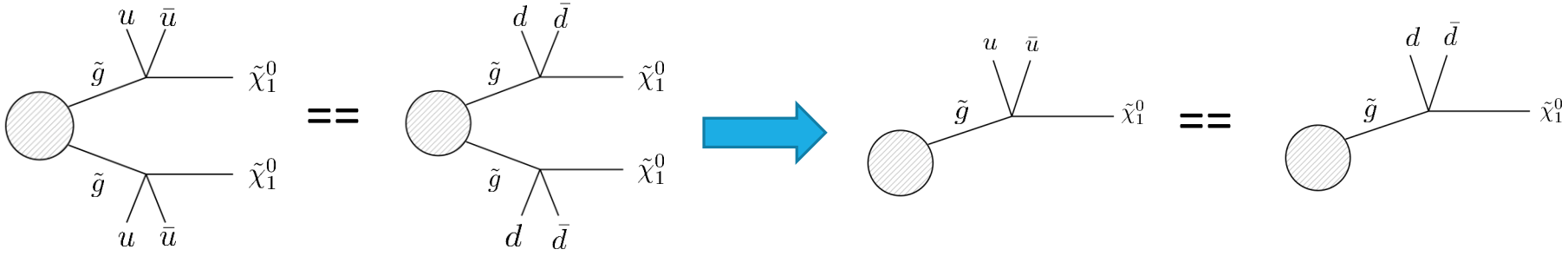
Element

Branch

Particle



Comparing elements: el1 == el2



Branch comparison:

- Is it a `wildcard`?: No
- `vertnumb`, `vertparts`: 1 ==1, 2 ==2
- Compare final state `particles, now` by label: ‘q’, ‘q’ == ‘q’, ‘q’
- Compare `BSM particles` by Z2-parity and mass: `gluino.Z2parity == gluino.Z2parity, n1.Z2parity == n1.Z2parity`, same for mass
- Compare `BSM particles` by Z2parity, color dimension, electric charge

Comparison of elements I: el1 == el2
Used for grouping elements
When comparing elements `Particle class` objects are compared
now compare the attribute `label` of `particle objects`

Comparing elements: el1.particlesMatch(el2)

Used for comparison theory element and database element

Different than '==' because of inclusive labels (like jet) and sorting of element

`el1.particlesMatch(el2)` → `el1.branches[0].particlesMatch(el2.branches[0])`

Branch particles match:

- `vertnumb`, `vertparts`
- Compare particles with `simparticles`:
  - replace inclusive labels represented by `ParticleList objects` with `particles` contained in it, e.g. `jet` → `[u,d,c,s, pip, piz, gluon +` conjugates`]`
  - Use `itertools` to create all combinations arising from different sorting
  - Compare lists of `particle objects`
- Compare last `BSM particle` of each branch by quantum numbers

Missing topologies

= All elements that did not match an experimental result

When reweighting factors are included in database, missing topologies do not get reweighted

Use approximate exponential functions

Approximate estimation of $\gamma\beta$ ok because missing topologies to give overview over what has not been covered

Also include displaced topologies: $F_{displaced} = 1 - F_{prompt} - F_{long}$

Database example MET

```
txName: T2bb
constraint: [[['b']],[['b']]]
condition: None
conditionDescription: None
figureUrl: http://cms-results.web.cern.ch/cms-results/public-results/preliminary-results/SUS-16-016/CMS-PAS-SUS-16-016_Figure_005-c.png
source: CMS
validated: True
axes: [[x, y], [x, y]]
upperLimits: [[[[3.0000E+02*GeV,0.0000E+00*GeV],[3.0000E+02*GeV,0.0000E+00*GeV]],1.0613E-01*pb],
[[[3.0000E+02*GeV,1.0000E+02*GeV],[3.0000E+02*GeV,1.0000E+02*GeV]],1.5800E-01*pb],
[[[3.0000E+02*GeV,2.0000E+02*GeV],[3.0000E+02*GeV,2.0000E+02*GeV]],5.5344E-01*pb],
[[[3.0000E+02*GeV,2.5000E+02*GeV],[3.0000E+02*GeV,2.5000E+02*GeV]],1.4901E+00*pb],
[[[3.0000E+02*GeV,2.7500E+02*GeV],[3.0000E+02*GeV,2.7500E+02*GeV]],2.5446E+00*pb],
[[[3.2500E+02*GeV,2.5000E+02*GeV],[3.2500E+02*GeV,2.5000E+02*GeV]],8.7752E-01*pb],
[[[3.2500E+02*GeV,2.7500E+02*GeV],[3.2500E+02*GeV,2.7500E+02*GeV]],1.0126E+00*pb],
```

https://github.com/SModelS/smodels/blob/master/smodels-database/13TeV/CMS/CMS-PAS-SUS-16-016/data/T2bb.txt

Database example + new features

Results for signatures beyond missing energy often in form of inclusive results → introduce * to represent the inclusiveness

```
txName: THSCPM2
constraint: [[*],[]]
condition: None
conditionDescription: None
susyProcess: pp --> lsp chargino^pm_1, chargino^pm_1 lsp --> chargino^pm_1 lsp
figureUrl: None
dataUrl: None
source: SModelS
validated: N/A
axes: [['*'], [x]]
finalState: ['MET', 'HSCP']
efficiencyMap: [[['*',[1.5000E+02*GeV]],0.12396],
[['*',[1.0000E+02*GeV]],0.058038]]
```

Considered the same when compared to same type

(not used)

Final states

https://github.com/SModelS/smodels/blob/hscp-particleClass/smodels-database/13TeV/CMS/CMS-PAS-EXO-16-036-eff/c000/THSCPM2.txt

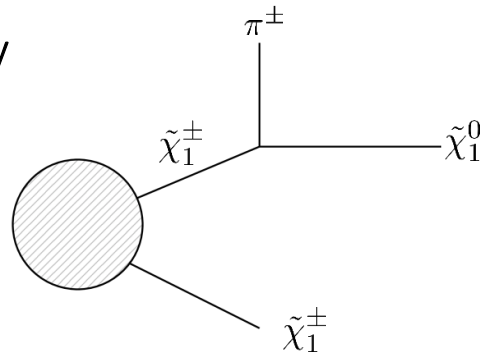
Database

```
txName: THSCPM4
constraint: [[*],[['*']]]
condition: None
conditionDescription: None
susyProcess: pp --> squark squark, squark --> quark chargino_1 (quark lsp)
figureUrl: None
dataUrl: None
source: SModelS
validated: N/A
axes: [['*'], [x, y]]
finalState: ['MET', 'HSCP']
efficiencyMap: [[['*',[2.3500E+02*GeV,1.0000E+02*GeV]],0.10007],
[['*',[5.1000E+03*GeV,1.0000E+02*GeV]],0.0],
[['*',[2.5100E+02*GeV,1.5000E+02*GeV]],0.16765],
```

(not used)

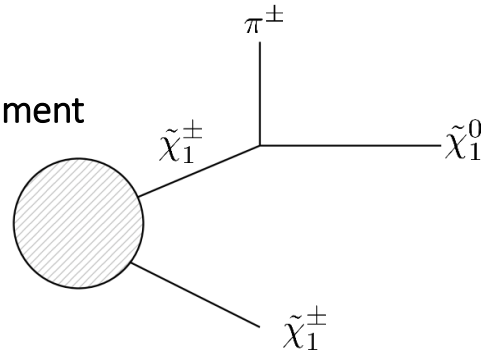
Example

theory




```
BSMparticles = [c1, n1]
BSMparticles = [c1]
```

experiment




```
BSMparticles = [anyBSM, MET]
BSMparticles = [HSCPp/m]
```

New features on theory side

Wildcards for Branch, Particle:

- Handle inclusive constraints from database
- Always True when compared to object of same type
- [*] → Branch wildcard: `BranchWildcard()`
- '*' → Particle wildcard: `ParticleWildcard(label='*',Z2parity='even')`
- Examples:
  - [*] is equal to [ ] or [[e-, e+]] or [[q], [q,q]] or …
  - '*' is equal to e, Z, …

New features on theory side

Final states:

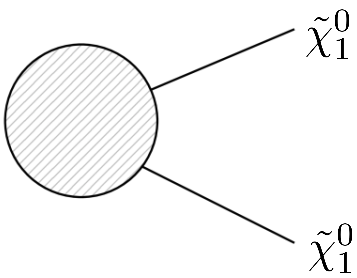
- Add a final state particle to BSM particles as specified in database

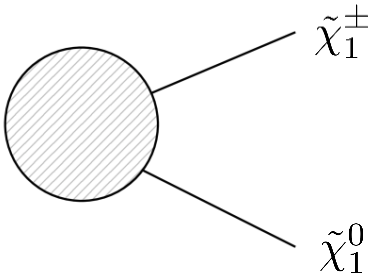
```
MET = Particle(
label='MET',
Z2parity ='odd‘,
eCharge = 0,
colordim = 0)
```

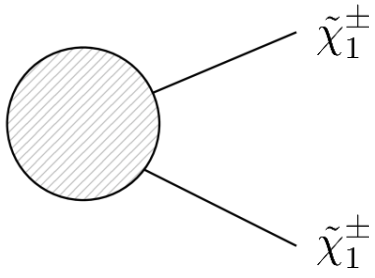
```
HSCPp =
Particle(
label='HSCP+',
Z2parity ='odd‘,
eCharge = +1,
colordim = 0)
```

- Intermediate BSM particles on experimental side:
  `ParticleWildcard(label='anyBSM',Z2parity='odd‘)`

Differentiation between MET and HSCP







- All of these elements are [[ ], [ ]]
- In the database:
  - Final states: ‘HSCP’, ‘MET’
  - If not defined in database → ‘MET’

Missing topologies

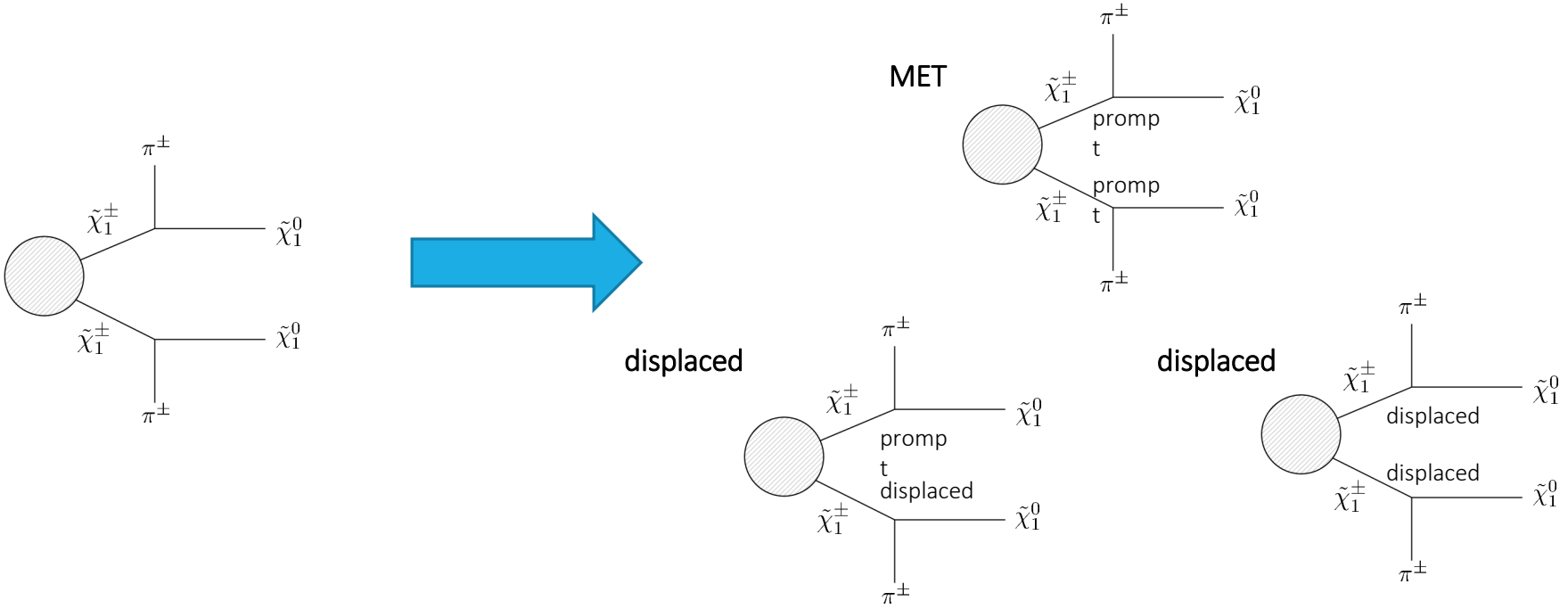
Procedure: for Branch, Particle:

- For very element produce new elements according to all possible combinations of final states/ kinds of decays
- Reweight with exponential functions
- Label elements to categorize in MET, long-lived, displaced

Naming:

- Displaced: as soon as one displaced decay happened
- MET: all decays prompt and last BSM particle stable and neutral
- Long-lived: all decays prompt and last BSM particle stable and charged

Missing topologies



Probabilities for ‘subelements‘

$$
\begin{aligned}
P_{non-prompt} &= (1 - F_{prompt} - F_{long})^2 * \tilde{F}_{long}^2 + 2 * (1 - F_{prompt} - F_{long}) * F_{prompt} * \tilde{F}_{long}^2 \\
&= (1 - F_{prompt} - F_{long}) * ((1 - F_{prompt} - F_{long}) + 2 * F_{prompt}) * \tilde{F}_{long}^2 \\
&= (1 - F_{prompt} - F_{long}) * ((1 + F_{prompt} - F_{long})) * \tilde{F}_{long}^2 \\
&= ((1 - F_{long})^2 - F_{prompt}^2) * \tilde{F}_{long}^2
\end{aligned}
$$